ISSN 2321-807X
Volume 12 Number17
Journal of Advances in chemistry

A SURVEY ON ANT COLONY OPTIMIZATION ALGORITHM

Mr.E.Balraj,Asst.Professor/IT,Mrs.R.Sujatha,Asst.Professor/IT

M.Kumarasamy College of Engineering,Karur,India

balraje.it@mkce.ac.in

M.Kumarasamy College of Engineering,Karur,India

sujathar.it@mkce.ac.in

ABSTRACT

A novel Ant Colony Optimization algorithm (ACO) combined for the hierarchical multi- label classification problem of protein function prediction. This kind of problem is mainly focused on biometric area, given the large increase in the number of uncharacterized proteins available for analysis and the importance of determining their functions in order to improve the current biological knowledge. Because it is known that a protein can perform more than one function and many protein functional-definition schemes are organized in a hierarchical structure, the classification problem in this case is an instance of a hierarchical multi-label problem. In this classification method, each class might have multiple class labels and class labels are represented in a hierarchical structure—either a tree or a directed acyclic graph (DAG) structure. A more difficult problem than conventional flat classification in this approach, given that the classification algorithm has to take into account hierarchical relationships between class labels and be able to predict multiple class labels for the same example. The proposed ACO algorithm discovers an ordered list of hierarchical multi-label classification rules.

Indexing terms/Keywords

ACO – Ant Colony Optimization, DAG-directed Acyclic graph, MuLAM (Multi-Label Ant-Miner), *h*Ant-Miner (Hierarchical Classification Ant-Miner),

Academic Discipline And Sub-Disciplines

Datamining

SUBJECT CLASSIFICATION

Optimization

TYPE (METHOD/APPROACH)

Survey

INTRODUCTION

Classification is one of the important d a t a mining tasks. The main objective of this is to learn a relationship between input values and a desired output. A set of examples defined by a classification problem, where each example is explained by predictor attributes and associated with a class attribute. It consists of t w o phases. F irst phase consists of g i v e n a labeled data set—a data set consisting of examples with a known class value (label) as an input, a classification model that represents the relationship between predictor and class attribute values is built. The second phase, the classification model is used to classify unknown examples— examples with unknown class value.

Most of the classification algorithms are discussed in the previous algorithms, each example is associated with only one class value or label and class values are unrelated—i.e. there are no relationships between different class values. The above said classification problems are usually referred to as flat (non-hierarchical) single-label problems. The main problem of hierarchical multi label classifications are, examples

may be associated to multiple class values at the same time and the class values are organized in a hierarchical structure (e.g. a tree or a directed acyclic graph structure). According to the data mining perspective, hierarchical multi-label classification is more challenging than flat single-label classification. Most difficult task of hierarchical multi label classification is to discriminate between classes represented by nodes at the bottom of the hierarchy than classes represented by nodes at the top of the hierarchy, since the number of examples per class tends to be smaller at lower levels of the hierarchy as opposed to top levels of the hierarchy. Another problem is, class predictions must satisfy hierarchical parent-child relationships, since an example associated with a class is automatically associated with all its ancestors classes. Final problem is multiple unrelated classes-classes which are not involved in ancestor/descendant relationship may be predicted at the same time..

There has been an increasing interest in hierarchical classification, where in general early applications are found in text classification and recently in protein function prediction .The latter is a very active research field, given the large increase in the number of uncharacterized proteins available for analysis and the importance of determining their functions in order to improve the current biological knowledge. It is important to emphasize that in

this context, comprehensible classification models which can be validated by the user are preferred in order to provide useful insights about the correlation of protein features and their functions. Concerning the problem of protein function prediction, the focus of we, an example to be classified corresponds to a protein, predictor attributes correspond to different protein features and the classes correspond to different functions that a protein can perform. Since it is known that a protein can perform more than one function and function definitions are organized in a hierarchical structure (e.g. FunCat and Gene Ontology protein functional-definition schemes), the classification problem in this case is an instance of a hierarchical multi-label problem.

2.LITERATURE SURVEY

2.1 Ant Colony Optimization

Ant Colony Optimization algorithms simulate the behavior of real ants using a colony of artificial ants, which cooperate in finding good solutions to optimization problems. Every artificial ant, representing a simple agent, builds candidate solutions to the problem at hand and communicates indirectly with other artificial ants by means of pheromone values. At the same time that ants perform a global search for new solutions, the search is guided to better regions of the search space based on the quality of solutions found so far. The algorithm converges to good solutions as a result of the collaborative interaction among the ants; an ant probabilistic chooses a trail to follow based on heuristic information and pheromone values, deposited by previous ants. The interactive process of building candidate solutions and updating pheromone values allows an ACO algorithm to converge

2.2 MuLAM Optimization

It proposed a new ACO algorithm, named MuLAM (Multi-Label Ant-Miner), for discovering multi-label classification rules. In essence, MuLAM differs from the original Ant-Miner in three aspects, as follows. Firstly, a classification rule can predict one or more class attributes, as in multi-label classification problems an example can belong to more than one class. Secondly, each iteration of MuLAM creates a set of rules instead of a single rule as in the original Ant-Miner. Thirdly, it uses a pheromone matrix for each class value and pheromone updates only occur on the matrix of the class values that are present in the consequent of a rule. In order to cope with multi-label data, MuLAM employs a criterion to decide whether one or more.

2.3 hAnt Miner

This Algorithm proposed an extension of the flat classification Ant-Miner algorithm tailored for hierarchical classification problems, named *h*Ant-Miner (Hierarchical Classification Ant-Miner), employing a hierarchical rule evaluation measure to guide pheromone updating, a heuristic information adapted for hierarchical classification, as We as an extended rule representation to allow hierarchically related classes in the consequent of a rule. However, *h*Ant-Miner cannot cope with hierarchical multi-label problems, where an example can be assigned to multiple classes that are not ancestor/descendant of each other.

3 .EXISTING SYSTEM

*h*Ant-Miner algorithm is the discovery of hierarchical classification rules in the form *IF antecedent THEN consequent*. The antecedent of a rule is composed by a conjunction of conditions based on predictor attribute values (e.g. *length* > 25 AND *IPR00023* = *yes*) while the consequent of a rule is composed by a set of class labels in potentially different levels of the class hierarchy respecting ancestor/decendant class relationships (e.g., GO:0005216, GO:0005244—where GO:0005244 is a subclass of GO:000-5216). This algorithm divides the rule construction process into two different ant colonies, one colony for creating antecedent of rules and one colony for creating consequent of rules, and the two colonies work in a cooperative fashion.

After the rule construction procedure has finished, the rules constructed by the ants are pruned to remove irrelevant terms (attribute-value conditions) from their antecedent— which can be regarded as a local search operator—and class labels from their consequent. Then, pheromone levels are updated using the best rule (based on a quality measure *Q*) of the current iteration and the best-so-far rule (across all iterations) is stored. The rule construction procedure is repeated until a user-specified number of iterations has been reached, or the best-so-far rule is exactly the same in a predefined number of previous iterations. The best-so-far rule found is added to the rule list and the covered training examples—i.e. examples that satisfy the rule's antecedent conditions—are removed from the training set.

Overall, *h*Ant-Miner can be regarded as a memetic algorithm, in the sense that it combines conventional concepts and methods of the ACO meta heuristic with concepts and methods of conventional rule induction algorithms (e.g. the sequential covering and rule pruning procedures), as discussed earlier.

According to discover a list of classification rules, a sequential covering approach is employed to cover all (or almost all) training examples. Algorithm 1 presents a high- level pseudo code of the sequential covering procedure employed in *h*Ant-Miner.

Algorithm 1

input : *training examples* output: *discovered rule list*

```
1  begin
2  training set ←all training examples;
3  rule list ←0/ ;
4  while |training set| > max uncovered examples do
5  rulebest ←0/ ;
6  i←1;
7  repeat
8  rulecurrent ←0/ ;
9  for j←1 to colony size do
10  // use separate ant colonies for antecedent and consequent construction
11  rulej ←CreateAntecedent()+CreateConsequent();
12  // applies a local search operator
13  Prune(rulej);
14  // updates the reference to the best rule of the iteration
15  if Q(rulej) > Q(rulecurrent) then
16  rulecurrent ←rulej ;
17  end
18  j← j+1;
19  end
20  UpdatePheromones(rulecurrent );
21  if Q(rulecurrent) > Q(rulebest) then
22  rulebest ←rulecurrent ;
23  end
24  i←i+1;
25  until i ≥ max number iterations OR RuleConvergence() ;
26  rule list ←rule list +rulebest ;
27 training set ←training set −Covered(rulebest ,training set);
28  end
29  return rule l
```

4.PROBLEM DESCRIPTION

While analyzing *h*Ant-Miner, We have identified the following limitations.

The main drawback of the hAntMiner is heuristic information, which involves a measure of entropy, used in *h*Ant-Miner is not very suitable for hierarchical classification—i.e. it will not consider for identifying the hierarchical relationships between classes. Even though *h*Ant-Miner's entropy measure is calculated throughout all labels of the class hierarchy (apart from the root label), each class label is evaluated individually without considering parent-child relationships between class labels.

Another drawback is, the measurement of rule quality is prone to over fitting. Because only the examples covered by the rule are considered in the rule evaluation, rules with a small coverage are favoured over more generic rules. Let We consider the example, the class label 1.2.1 with 20 examples and two rules that have class 1.2.1 as the most specific class label in their consequent: *rule*1 covering correctly 5 examples out of a total of 5 covered and *rule*2 covering correctly 19 examples out of a total of 20 covered. According to this situation, *rule*1 would have a higher quality, because all the examples covered by the rule are correctly classified, than *rule*2, which misclassifies one ex- ample, though *rule*2 covers all but one examples belonging to class 1.2.1.Important issue that the rule quality mea-sure of *h*Ant-Miner could be easily modified to avoid over- fitting by evaluating a rule considering all the examples of its most specific class. The drawback of this approach is that it favors rules predicting class labels at the top of the hierarchy, since the numbers of examples per

class are greater at top class levels. This could potentially prevent the discovery of rules predicting more specific class labels given that the examples covered by a rule are removed from the training set—indeed; this problem was observed in some preliminary experiments.

At last, I t does not support multi-label data since a single path in the consequent construction graph corresponds to the consequent of a rule. If We take protein function prediction, where it is known that a protein can perform more than one function. So it also considered as one of the important issue .

5.PROPOSED WORK

A new hierarchical multi-label ant colony classification algorithm, named *hm*Ant-Miner (Hierarchical Multi-Label Classification Ant-Miner) is developed to overcome the aforementioned limitations. E v e n *hm*Ant-Miner shares the same underlying procedure of the *h*Ant-Miner algorithm as We have seen, it differs from *h*Ant-Miner in the following aspects:

- The consequent of a rule is evaluated using a deterministic procedure based on the examples covered by the rule, allowing the creation of rules that can predict more than one class label at the same time (multi-label rules). Therefore, *hm*Ant-Miner uses a single construction graph in order to create a rule—only the antecedent is rep- resented in the construction graph;
- Euclidean distance is used to define the heuristic function, where each example is represented by a vector of

  a vector of class membership values in the Euclidean space. Instead of using entropy in hAntMiner. We can use distrance measure to help us to identify possible values and to take into account the relationship between class labels given that examples belonging to related (ancestor/decendant) class labels will be more similar than examples belonging to unrelated class labels, This concept is inspired from CLUS –HMS algorithm for hierarchical multi-label classification, It is based on the paradigm of decicion tree induction, rather than rule induction.
- A distance based measure can be used to evaluate the rule quality, which is a more suitable evaluation measure for hierarchical multi-label problems;
- Rule pruning procedure is not applied to the consequent of a rule. It is (re-)calculated when its antecedent is modified during pruning, since the set of covered examples might have changed

5.1 The Consequent Rule Construction

The consequent of rule is calculated in h_mAnt Miner by using the following deterministic procedure.

$$\text{Consequent } r,i = \frac{|SI_r \ \& \text{ label } i|}{|SI_r|}$$

$|SI_r \ \&$ label i $|$ - the number of examples covered byrule *r* that belong to the *i*-th class of the class hierarchy(*labeli*)

Sr - covered by a rule *r*

5.2 The distance based Heuristic Information

Heuristic information in *h*Ant-Miner involves a measure of entropy, as in the original Ant-Miner. The entropy characterizes the homogeneity . The entropy characterizes the homogeneity of a collection of examples related to the class attribute values, giving a notion of (im-)purity of the class values' distribution. The more examples of the same class the lower the value of entropy will be and the 'purest' is the collection of examples. It should be noted that in all calculations involving entropy, the different class labels (values) are independently evaluated—i.e. no relationship between class labels is taken into account. In the case of Ant-Miner, which is applied to flat classification problems, the use of the entropy measure does not present a limitation, since there is no relationship between class labels. On the other hand, the same cannot be said for *h*Ant-Miner, which aims at extracting hierarchical classification rules, derived from data where the class labels are organized in a hierarchical structure

To illustrate the limitation of the entropy measure when used in hierarchical problems, let us consider the following example. Given a tree-structured class hierarchy, where labels {1, 2, 3} are children of the root label and labels {2.1, 2.2} are children of the '2' label and each class label has 10 examples. Although the entropy is calculated according to Equation(4)—across all class labels, the hierarchical relationships are not taken into account. Therefore, the entropy of a hypothetical term '*IPR00023* = *yes*'which is present in10 examples of class '1' and in 10 examples of class '3' would be the same as of a hypothetical term'*IPR00023* = *no*' which is present in 10 examples of class'2' and in 10 examples of class '2.1'. The drawback in this case is that it is known that class labels '2' and '2.1' are more similar than class labels '1' and '3'. Hence, it would be expected/desired that the entropy measure (or an alternative heuristic information) exploit hierarchical relationships in order to better reflect the quality of each term in the case of hierarchical classification problems. Intuitively this becomes even more important when dealing with bigger (in terms of number of class labels and depth) hierarchical structures. It should be noted that several Ant-Miner variations—as dis- cussed have used a heuristic information based on the relatively frequency of the class predicted by the rule (or the majority class) among all the

examples that have a particular term, which would also present the above limitation.

*hm*Ant-Miner employs a distance-based heuristic information, which directly incorporates information from the class hierarchy. More precisely,the heuristic information of a term corresponds to the variance of the set of examples covered by the term (the set of examples that satisfy the condition represented by the term). In order to calculate the variance, the class labels of each example are represented by a numeric vector of length *m* (where *m* is the number of class labels of the hierarchy without considering the root label). The *i*-th component of the class label vector of an example is equal to 0 or 1 if the correspondent class label is absent or present, respectively. The distance betWeen class label vectors is defined as the Weighted Euclidean distance, given by

$$distance(v1, v2) = \sqrt{\sum_{i=1}^{m} w(l_i).(v_{1,i} - v_{2,i})}_2$$

where *w*(*li*) is the Weight associated with the *i*-th class label, v1,*i* and v2,*i* are the values of the *i*-th component of the class label vectors v1 and v2 , respectively. Then, the variance of a set of examples is defined as the averaged squared distance between each example's class label vector and the set's mean class vector, given by

$$\text{Variance(Sr)} = \frac{\sum_{k=1}^{|Sr|} distance(Vk,V)}{|Sr|}$$

where *Sr* is the set of examples covered by a term *T* and v is the set's mean class label vector. Finally, the heuristic information of a term *T* is given by

$$nr = \frac{variance_{max} - variance(s_r)}{variance_{max}}$$

where *variancemax* is defined as the sum of the worst and best variance values observed across all terms in order to assign values greater than zero to the worst terms, which other- wise would avoid them to be selected by an ant. Note that the heuristic value is normalized so the smaller the value of the variance of a term *T* the greater its heuristic value becomes. This is analogous to the use of the entropy measure in Ant-Miner and *h*Ant-Miner, where smaller values are preferred over bigger values since they correspond to a more homogeneous partition (where the great majority of examples belong to the same class).

5.3 Modified Rule Pruning

Proposed algorithm does not employ a second colony in order to consequent of rules construction. So the rule pruning procedure is simplified as follows. Every time rule is submitted to a removal process of its antecedent's last term and has its consequent re-calculated, because the set of covered examples could change after the removal of the term.

This kind of removal process is repeated until the quality of the rule decreases when its last term is removed or the rule has only one term left in the antecedent.

Let us consider the $rule_{current}$ be the rule undergoing the pruning - is considered the best rule at the beginning of the pruning procedure. Every iteration of the pruning procedure, a candidate rule $rule_i$ is created by removing the last term of the antecedent of

the current best $rule_{best}$ and the consequent of *rulei* is computed according to Subsection 5.1. Then, the

quality measure *qi* for *rulei* is computed. Let We compare the values of q i & qbest , If the quality measure *qi* is higher than the current best quality *qbest* ,$rule_i$ substitutes $rule_{best}$, completing an iteration of the pruning procedure. This procedure is repeated until $rule_{best}$ has just one term left on its antecedent or a candidate rule $rule_i$ does not improve the quality over $rule_{best}$ (i.e. $q_{best} > q_i$).

6. EXPERIMENTAL RESULT

There are two kinds of biometrics datasets has bee used for this proposed algorithm.

1. Gene Ontology dataset
2. Fun cat dataset

Table I Input dataset [Funcat]

dataset	FunCat			
	|training|	|test|	|attributes|	|classes|
cellcycle	2476	1281	77	500
desire	2450	1275	63	500
eisen	1587	837	79	462
expr	2488	1291	551	500
gasch1	2480	1284	173	500
pheno	1009	582	69	456
seq	2580	1339	478	500
spo	2437	1266	80	500

Table II Input dataset [Funcat]

dataset	Gene Ontology			
	|training|	|test|	|attributes|	|classes|
cellcycle	2473	1278	77	4126
desire	2447	1272	63	4120
eisen	1583	835	79	3574
expr	2485	1288	551	4132
gasch1	2477	1281	173	4126
pheno	1005	581	69	3128
seq	2568	1332	478	4134
spo	2434	1263	80	4120

Table III Average number of class labels in the hierarchy and the average class labels per example

	Fun cat	Gene Ontology
Average number of class labels	489	3932
Average labels per example	8.5	34.2

Table IV User Defined Parameters used by our dataset.

Table IV User Defined Parameters used by our dataset

Parameter	Description	Value
max uncovered examples	maximum number of uncovered examples	10
max number iterations	maximum number of iterations	1500
rule convergence	number of iterations used to test the rule convergence	10
min examples covered	minimum number of covered examples per rule	10
colony size	number of ants per iteration	30

7.CONCLUSION

The proposed paper presents a novelant colony algorithm tailored for hierarchical multi-label classification, named *hm*Ant-Miner (Hierarchical Multi-Label Classification Ant-Miner). Extending on the ideas of our previous hierarchical classification *h*Ant-Miner algorithm, *hm*Ant-Miner discovers a single global classification model, in the form of an ordered list of *IF-THEN* classification rules, which can predict all class labels from a class hierarchy at once, and examples may be assigned to multiple unrelated class labels. On account of the information from the class hierarchy, *hm*Ant-Miner employs a distance-based measure in the dynamic discretization procedure of continuous attributes and as heuristic information in the ACO construction graph. Because of that, the entropy measure used in *h*Ant-Miner is replaced by the distance measure in *hm*Ant-Miner, which is a more suitable measure for hierarchical multi-label classification.

Our proposed work have conducted experiments comparing *hm*Ant-Miner against state-of-the-art decision tree induction algorithms for Hierarchical multi-label classification with most challenging sixteen bioinformatics data sets involving the prediction of protein function, with large numbers of predictor attributes and large numbers of class labels to be predicted. Class hierarchies Were used in the experiments are represented in a tree (where a class label has a single parent, apart from the root label) or in a directed acyclic graph (where a class label can have multiple parents, apart from the root label) forms. We assure that *hm*Ant-Miner is most competitive in term of both predictive accuracy and simplicity We regard these results promising, given that *hm*Ant-Miner is the first ACO algorithm tailored for hierarchical multi-label classification, to the best of our knowledge.

REFERENCES

1. Abdul Rauf Baig (2013) Correlation as a Heuristic for Accurate and Comprehensible Ant Colony Optimization Based Classifiers. IN: IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 17, NO. 5, OCTOBER 2013.
2. Otero F, Freitas A, Johnson C (2009) A Hierarchical Classification Ant Colony Algorithm for Predicting Gene Ontology Terms. In: Proceedings of the 7th European Conference on Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics (EvoBio 2009), LNCS 5483, Springer, pp 68–79
3. Consortium TGO (2008) Gene ontology: tool for the unification of biology. Nature Genetics 25:25– 29. Otero F, Freitas A, Johnson C (2009) Handling contin- uous attributes in ant colony classification algorithms.In: Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Data Mining (CIDM-2009), IEEE, pp 225–231
4. Alves R, Delgado M, Freitas A (2008) Multi-label hierarchical classification of protein functions with artificial immune systems. In: Advances in Bioinformatics and Computational Biology (Proc. BSB-2008).
5. Bi R, Zhou Y, Lu F, Wang W (2007) Predicting Gene Ontology functions based on support vector machines and statistical significance estimation Neuro Computing 70.
6. Blockeel H, Schietgat L, Struyf J, Dz˘eroski S, Clare A (2006) Decision Trees for Hierarchical Multilabel Classification: A Case Study in Functional Genomics. In:PKDD-2006, LNAI 4213, pp 18–29
7. Barutcuoglu Z, Schapire R, Troyanskaya O (2006) Hierarchical multi-label prediction of gene function. Bioinformatics 22(7)

8. Chan A, Freitas A (2006) A new ant colony algorithm for multi-label classification with applications in bioinformatics. In: Proc. Genetic and Evolutionary Computation Conference (GECCO-2006), pp 27–34
9. Clare A, Karwath A, Ougham H, King R (2006) Functional bioinformatics for *Arabidopsis thailana*. Bioinformatics 22(9):1130–1136
10. Rousu J,Saunders C, Szedmak S, ShaWe-Tylor J (2006) Kernel Based learning of Hierarchical Multilevel classification Models. Journal of Machine learning Research pp1601-1626
11. Ruepp A, Zollner A, Maier D, Albermann K, Hani J, Mokrejs M, Tetko I, Guldener U, Mannhaupt G, Munsterkotter M, MeWes H (2004) The FunCat, a func- tional annotation scheme for systematic classification of proteins from whole genomes. Nucleic Acid Research 32(18):5539–5545
12. Cesa-Bianchi N, Zaniboni L, Collins M (2004) Incremental algorithms for hierarchical classification. Journal of Machine Learning Research pp 31–54
13. Blockeel H, Bruynooghe M, Dz˘eroski S, Ramon J, Struyf J (2002) Hierarchical multi classification. In: Dz˘eroski S, Raedt LD, Wrobel S (eds) Proceedings of the First SIGKDD Workshop on Multi-Relational Data Mining (MRDM 2002), University of Alberta, Edmon- ton, Canada, pp 21– 35
14. Parpinelli R, Lopes H, Freitas A (2002) Data mining with an ant colony optimization algorithm. IEEE

    Transactions on Evolutionary Computation 6(4):321–332.
15. Mohanaprabha G, Balraj E, (2014) A hm Ant-miner using evolutionary algorithm.International Journal

    of Innovative Research in Science, Engineering and Technology 1687-1692.

Author' biography with Photo

Balraj E was born in Dharapuram In the year of 1989. He received his bachelors degree in Computer Science and Engineering in 2010 from Anna University. He received his Masters Degree in Computer Science and Engineering from Anna University. He joined as Assistant Professor in M.Kumarasamy College of Engineering, Karur and he is working in the same college till date. His area of of Interest is Open Source Software and Data Mining.

Sujatha R was born in Chennai in the year 1974. She received her Masters degree in Computer Applications in 2002 from University of Madras. She received her Masters degree in Computer Science and Engineering in 2008 from Anna University.She Joined has Assistant Professor in M.Kumarasamy College of Engineering , Karur and Is working in the same college till date. Her area of interest is open source software,Datamining and Sensor networks.